Inf Retrieval (2011) 14:337–353
DOI 10.1007/s10791-010-9161-5

WEB MINING FOR SEARCH

Sentiment classification: a lexical similarity based approach for extracting subjectivity in documents

Kiran Sarvabhotla · Prasad Pingali · Vasudeva Varma

Received: 29 April 2010 / Accepted: 17 December 2010 / Published online: 12 February 2011

Abstract With the growth of social media, document sentiment classification has become an active area of research in this decade. It can be viewed as a special case of topical classification applied only to subjective portions of a document (sources of sentiment). Hence, the key task in document sentiment classification is extracting subjectivity. Existing approaches to extract subjectivity rely heavily on linguistic resources such as sentiment lexicons and complex supervised patterns based on part-of-speech (POS) information. This makes the task of subjective feature extraction complex and resource dependent. In this work, we try to minimize the dependency on linguistic resources in sentiment classification. We propose a simple and statistical methodology called review summary (RSUMM) and use it in combination with well-known feature selection methods to extract subjectivity. Our experimental results on a movie review dataset prove the effectiveness of the proposed methodology.

Keywords Social media · Sentiment classification · Subjectivity · Linguistic resources · RSUMM

1 Introduction

The textual information on the web can be broadly categorized into two: facts and opinions (Liu 2010; Pang and Lee 2008). Until the early part of this decade, most of the research work in the areas of natural language processing (NLP), text mining, and information retrieval (IR) focused on factual information. With the advent of customer reviews, blogs, and the growth of e-commerce in this decade, user-generated content has grown rapidly on the web (Liu 2010). It has an inherent property called *sentiment*. These sentiments are playing a prominent role in people's decision-making processes (Gretzel and Yoo 2008). Analyzing and predicting their polarity has received much attention in the research

K. Sarvabhotla (✉) · P. Pingali · V. Varma
Search and Information Extraction Lab, International Institute of Information Technology, Hyderabad, India
e-mail: kiransarv@research.iiit.ac.in

community and among market analysts for its potential business applications. Hence, sentiment classification has become one of the hot topics of research in this decade.

Sentiment analysis or classification predicts the polarity of a given text unit. The text unit can be a word, phrase, sentence or document. The polarity is predicted on either a binary[1] or multivariant scale.[2] The problem of sentiment classification[3] can be viewed as a special case of topical classification applied only to subjective portions of a document. In other words, topical classification focuses on keywords, whereas sentiment classification focuses on subjectivity in the text (Pang et al. 2002). Hence, the key task in sentiment classification is extracting the subjective portions from a document. In this work, we apply supervised learning approaches to classify the overall sentiment of a document[4] on a binary scale. We focus more on the aspect of extracting subjective features, existing approaches for doing it, problems with them, and present solutions.

Existing approaches in subjective feature extraction rely heavily on linguistic resources. Popular among them are lexicons such as SentiWordNet, General Inquirer, and part-of-speech (POS) tagger (Baccianella et al. 2009; Hu and Liu 2004; Matsumoto et al. 2005; Turney 2002). Lexicons are very generic and cannot capture subtle variations in sentiment expression from context to context and from domain to domain. Using POS tagger, researchers frame patterns that are assumed to be subjective based on POS information. The POS patterns vary from simple noun phrases (NP) to verb phrases (VP) and very complex patterns.[5] The text units that match these patterns in a document are extracted. There are also other techniques for extracting subjectivity, such as using WordNet, appraisal adjectives, and dependency parsing (Mullen and Collier 2004; Whitelaw et al. 2005; Matsumoto et al. 2005).

With the usage of linguistic resources and complex patterns, the task of subjective feature extraction has become more complex and resource dependent. As regional language content is growing on the web gradually, extending resource-based approaches for analyzing sentiments across several languages is a tedious job. It requires a lot of human effort to build such linguistic resources in each language. Also, inadequate availability of resources in a language should not prevent researchers from conducting experiments on analyzing sentiments. To make the task of sentiment classification more feasible, we need approaches that minimize the use of linguistic resources in subjective feature extraction.

We attempt to address the problem of resource dependency in sentiment classification by assuming that the entire document does not contain subjective information. The basis for our claim is manual observation of documents from different domains on the web. We have noticed many documents with less subjective content compared with total content. Hence, we can say that a sentiment-bearing document is a mixture of subjective and objective information, where the latter does not convey the feelings of the author. It has been proved in the literature that discarding the objective information enhances the performance of the sentiment classifier (Pang and Lee 2004). We propose a simple and statistical two-step filtering methodology for extracting subjective features from a document.

[1] Positive or negative.

[2] Grading reviews typically use a scale of 1–5, i.e., a starred rating (*).

[3] By sentiment classification we mean document sentiment classification.

[4] A document can be a review or a blog post.

[5] For example, NP, VP, JJ NN, RB JJ not NN, JJ JJ not NN, RB VB, NN JJ not NN, etc., where NN is a noun and RB stands for adverb, JJ stands for adjective, and VB stands for verb.

In the first step, we estimate the subjectivity of each sentence in a document. We use techniques very similar to the vector space model in IR. We call this method of estimating subjectivity as review summary (RSUMM). We define two metrics, the average document frequency (ADF) and the average subjective measure (ASM), and use them in RSUMM to estimate the subjectivity of each sentence. Our RSUMM is based on a lexical similarity model between two term vectors. It retains the most subjective sentences in a document, discarding the objective information. Thus, we obtain an excerpt that preserves the subjectivity at a level comparable to or better than the original document.

Later, we represent the document and its subjective excerpt as a feature vector to the classifier using *n*-gram models. In the next step, we apply two well-known feature selection methods, mutual information (MI) and Fisher discriminant ratio (FDR), on *n*-gram feature vectors to obtain the final feature set. We conduct experiments on a movie review dataset to validate our proposed two-step filtering methodology. Through our experimental results, we conclude that subjective feature extraction is possible with minimum usage of linguistic resources.

RSUMM was inspired from the work by Pang and Lee (2004). They established a strong relationship between sentence-level subjectivity detection and document-level sentiment classification. In their work, sentence-level subjectivity detection increased the performance of the sentiment classifier. They used a min-cut graph-based classification framework using individual and contextual penalties for each sentence to obtain a subjective excerpt of a document.

The rest of the paper is organized as follows. In Sect. 2, we discuss the work related to document sentiment classification, subjective feature extraction, and related existing techniques. In Sect. 3, we describe RSUMM, which estimates the subjectivity of each sentence in a document and optimizes the sentiment. In Sect. 4, we discuss how to represent text units as feature vectors to the classifier. We also discuss the feature selection methods for obtaining the final feature set. In Sect. 5, we describe our experimental activity, including description of the datasets, evaluation measures, and results. In Sect. 6, we discuss the performance of the sentiment classifier and present our observations. Finally, we conclude the paper by giving possible future directions in Sect. 7.

2 Related work

Sentiment classification dates back to the late 1990s (Hatzivassiloglou and McKeown 1997; Argamon et al. 1998; Kessler et al. 1997), but in the early part of this decade, it has become an important discipline in the areas of NLP, text mining, and information retrieval. The classification is done at several levels of text units. The text unit can be a word, phrase, sentence or document. In this section, we discuss more on the work related to document sentiment classification, subjective feature extraction using linguistic resources, and related resource- or language-independent approaches. Among the existing approaches in document-level classification, supervised learning methods[6] are popular among researchers to predict polarity. The movie review domain is popular among them (Pang et al. 2002; Mullen and Collier 2004; Baccianella et al. 2009; Matsumoto et al. 2005; Pang and Lee 2004; Beineke et al. 2004; Whitelaw et al. 2005). This may be due to the abundant availability of movie reviews on the web and their challenging nature (Turney 2002).

[6] Support vector machines (SVM), naive Bayes, and maximum entropy-based classification.

Document sentiment classification is typically composed of two steps: (1) extraction of subjective features from training data and their conversion to feature vectors, and (2) training of the classifier on the feature vectors and application of the classification to an unseen sample. Raw documents are preprocessed before extracting the subjective features. The preprocessing stage includes removal of hypertext markup language (HTML) tags from a document, tokenization, etc.

At the word level, Hatzivassiloglou and McKeown (1997) used conjunctive expressions to extract semantic polarities of words. The approach was based on linguistic constraints that "and" always conjuncts two words with the same orientation whereas "but" contradicts them. Turney proposed an unsupervised approach to predict the overall sentiment of a document (Turney 2002). The approach was based on pointwise mutual information (PMI) between a given phrase and the words "excellent" and "poor" to predict the semantic orientation. The phrases were extracted using Brill's tagger based on a set of POS patterns as mentioned in Sect. 1.

Pang et al., in 2002, experimented with several machine learning algorithms using unigrams, bigrams, POS information, and sentence position as features (Pang et al. 2002). They reported that support vector machines (SVM) with unigrams as features yielded the best results. They conducted experiments on a movie review dataset and concluded that machine learning techniques outperform human-produced baselines. They predicted the polarity of a review on a binary scale and reported an accuracy of 82.9%. Mullen and Collier (2004) used diverse information scores that assign value to each word or phrase using WordNet, topic proximity, and syntactic relations. They reported an accuracy of 86% on the same dataset.

Matsumoto et al. (2005) used word subsequence mining and dependency parsing for extracting subjective features. Their motive was to preserve word order and syntactic relations between words in sentences or clauses. They extracted clauses using a clausal extraction tool. They imposed some linguistic constraints on the extracted clauses based on POS information. They used *n*-gram features to represent each review by setting intuitive support thresholds for selecting a feature. They reported a maximum accuracy of 88.3% on the same movie review dataset.

Pang et al., in 2004, examined the relation between sentence-level subjectivity detection and document-level polarity classification (Pang and Lee 2004). They did not use any linguistic resource, instead training a naive Bayes classifier on an annotated subjective/objective collection. In addition, they incorporated contextual information and used a min-cut-based classification framework to obtain a subjective excerpt of a document. They reported an increase in accuracy of about 4.5% using the subjective excerpt compared with full review of a movie review dataset. Matsumoto et al. (2005) applied subsequence mining, dependency parsing techniques on the same dataset and reported an increase in accuracy of about 7% over Pang et al.'s accuracy values. However, their approach used linguistic resources such as clausal extraction tools, POS tagger, and dependency parsing.

Thet et al. (2008) conducted experiments on aspect-level sentiment classification of movie reviews. They used information extraction techniques such as entity extraction, co-referencing, and pronoun resolution to segment the text into sections, where a particular section focuses on a particular aspect of a product (movie). They predicted the sentiments of people towards crew (casts, directors) and the overall sentiment of a review. Baccianella et al. (2009) used lexicons such as General Inquirer and POS patterns for extracting subjective features in a document. They used the ε-support vector regression (ε-SVR) method to predict polarity. They proposed a new feature selection method called minimum variance (MV) to select relevant features. They also implemented a round-robin way of

selecting features to minimize the impact of dataset skew in their experiments. They conducted experiments on hotel reviews and graded them on an ordinal scale of one to five (starred rating).

There are approaches in the literature that minimize or do not use any resource in sentiment classification. Cui et al. (2006) used an n-gram model to represent each document as a feature vector and compared the performance of different classifiers. Their claim was that the existing work in sentiment classification focused on a small set of documents. They wanted to test the performance of different sentiment classifiers on a large dataset with statistical n-gram models. Hu et al. (2007) used information retrieval techniques based on a language model to predict orientation. They used a combination of Kullback–Leibler divergence and different smoothing techniques to predict polarity.

Raychev and Nakov (2009) used a different weighting scheme based on word position and its subjectivity for a naive Bayes classifier. They conducted experiments on a standard Internet Movie Database (IMDb) movie review dataset and reported an accuracy of 89.5% using their modified weighting scheme. In 2010, Li et al. (2010) proposed an approach that combines polarity shifting of sentences and document polarity classification. Polarity shifting means that the polarity of a sentence is different from the polarity expressed by the sum of the content words in it.[7] Their motive was that default polarity classification techniques could not capture subtle variations in the polarity of each sentence. Hence, they followed a metaclassifier approach, where they estimate the shift in polarity of each sentence and then predict the overall polarity of a document. They conducted experiments on different datasets with minimal use of linguistic resources.

There are also other topics of research in sentiment analysis that have become popular recently. Generally, sentiment analysis tasks are highly domain dependent. A classifier trained on one domain may not perform accurately on another (Tan et al. 2009; Aue and Gamon 2005). Hence, researchers are carrying out experiments to adapt classifiers for cross-domain sentiment classification. There are also other tasks such as utility of reviews, review spam detection, etc.

3 RSUMM

Our subjective feature extraction process has two steps: In the first step, we propose a simple and statistical methodology called RSUMM for estimating the subjectivity of each sentence in a document. We obtain a subjective excerpt of a document by discarding sentences that are objective. We represent the document and its subjective excerpt using an n-gram model. In the second step, we apply well-known feature selection methods on the resultant n-grams to obtain the final feature set. In this work, we focus more on estimating subjectivity and obtaining the subjective excerpt of a document.

Our RSUMM is similar to vector space modeling techniques used in information retrieval. We view each document d as a mixture of subjective and objective information, where the former conveys the feelings of the author on a particular topic and the latter is facts. We represent each sentence $s \in d$ as a vector of terms, $\bar{s} := (t_{s,1}, t_{s,2}, \ldots t_{s,n})$. We define two metrics, the average document frequency (ADF) and the average subjective measure (ASM), and derive the corresponding term vectors $\overline{adf}$ and $\overline{asm}$ for the respective collections. We then compute the lexical similarity between $\bar{s}$ and the vectors $\overline{adf}$, $\overline{asm}$, and

[7] The presence of negation, contrast transition, etc.

estimate its subjectivity using Jaccard similarity measure. We retain the top $X\%$ of sentences from each document d as its subjective excerpt.

3.1 Vector space model

The vector space model assigns weights to index terms (Baeza-Yates and Ribeiro-Neto 1999). It is widely used in information retrieval to determine the relevance of a document for a given query. Both document and query are represented as weighted vectors of terms, and these weights are used to compute the degree of similarity between the query and the document. The higher the degree of similarity, the more relevant the document is to the query.

Formal definition *Both query q and document d are represented as a weighted vector of terms. The query vector is defined as* $q := (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$, *and the document vector as* $d := (w_{1,d}, w_{2,d}, \ldots, w_{t,d})$, *where t is the total number of index terms.*

Then, the degree of similarity between the document d and the query q is the correlation between the two vectors. The correlation is quantified by a variety of similarity measures, for instance, by the cosine of the angle between the two vectors. The weighting measure used typically in vector space model is *tfidf*.

$$TFIDF(t, C) = tf(t, d) \times \log\left(\frac{N}{n}\right), \tag{1}$$

where $tf(t, d)$ denotes the frequency of the term in the given document d, N denotes the total number of documents in the collection C, and n denotes the number of documents containing the term t in C.

$$\begin{aligned} \cos\theta &= \frac{\bar{d}.\bar{q}}{\|\bar{d}\|\|\bar{q}\|} \\ &= \frac{\sum_{i=1}^{t} w_{i,d} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,d}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}. \end{aligned} \tag{2}$$

3.1.1 Average document frequency (ADF)

Document frequency is a widely used statistical measure in information retrieval to determine the importance of a term in a given collection. We use the ADF metric to represent the collection C_{pol}[8] as a vector of terms, $adf := (t_{adf,1}, t_{adf,2}, \ldots, t_{adf,n})$. Each term $t_{adf} \in adf$ has a document frequency greater than the average document frequency of the collection. This metric intuitively selects the most important features from a given collection and features specific to its domain.

$$ADF(C_{pol}) = \frac{\sum_{i=1}^{|V_{pol}|} df(t_i, c_{pol})}{|V_{pol}|}, \tag{3}$$

where $ADF(C_{pol})$ denotes the average document frequency of the collection C_{pol}, $|V_{pol}|$ denotes the total number of terms present in C_{pol}, and $df(t_i, c_{pol})$ denotes the document frequency of term t_i in the collection C_{pol}.

[8] An annotated collection of documents.

3.1.2 Average subjective measure (ASM)

We use the ASM metric to represent a collection C_{sub}[9] as a vector of terms. Each term in *asm* has subjective measure greater than the average subjective measure of the collection, $asm := (t_{asm,1}, t_{asm,2}, \ldots, t_{asm,n})$. This metric intuitively selects the most subjective features in a given annotated collection.

The subjective measure of a term, $t_i \in C_{sub}$, is calculated as follows:

$$\Phi(t_i, C_{sub}) = \frac{subj(t_i, C_{sub})}{obj(t_i, C_{sub}) + tot(C_{sub})}, \tag{4}$$

where $\Phi(t_i, C_{sub})$ denotes the subjective measure of $t_i \in C_{sub}$, $subj(t_i, C_{sub})$ denotes the frequency of term t_i in subjective instances of the annotated collection C_{sub}, $obj(t_i, C_{sub})$ denotes the frequency of term t_i in objective instances of the annotated collection C_{sub} (penalizing factor), and $tot(C_{sub})$ denotes the total number of instances in C_{sub} (normalizing factor).

The average subjective measure of a collection is calculated as follows:

$$ASM(C_{sub}) = \frac{\sum_{i=1}^{|V_{sub}|} \Phi(t_i, C_{sub})}{|V_{sub}|}, \tag{5}$$

where $ASM(C_{sub})$ is the average subjective measure of the collection C_{sub}, $|V_{sub}|$ denotes the total number of terms in C_{sub}, and $\Phi(t_i, C_{sub})$ denotes the subjective measure of term $t_i \in C_{sub}$ calculated using Eq. (4).

3.1.3 Final scoring

After obtaining both the *adf* and *asm* vectors using Eqs. (3) and (5), respectively, we compute the lexical similarity between each sentence vector s and the vectors *adf* and *asm*. In our work, we use raw terms rather than their weights. We use the Jaccard similarity measure to compute the similarity between two given vectors, as shown in Eq. (6)

$$\sigma(\bar{a}, \bar{b}) = \frac{n(\bar{a} \cap \bar{b})}{n(\bar{a} \cup \bar{b})}, \tag{6}$$

where $\sigma(\bar{a}, \bar{b})$ denotes the similarity score between the two vectors a and b, $n(\bar{a} \cap \bar{b})$ denotes the number of terms overlapping between vectors a and b, and $n(\bar{a} \cup \bar{b})$ denotes the total number of terms in both vectors.

The final score of a sentence s, $FS(s)$, is a combination of lexical similarity scores, as shown in Eq. (7). We rank sentences in document d in decreasing order of $FS(s)$ and retain the top $X\%$ of them. The first part of Eq. (7) computes the lexical similarity between a sentence and the most important features in a collection. Hence, the most informative sentence in a document gets high priority. The latter part of it computes the lexical similarity between a sentence and the most subjective terms. So, sentences that are more subjective are ranked higher.

$$FS(s) = \sigma(\overline{adf}, \bar{s}) + \sigma(\overline{asm}, \bar{s}). \tag{7}$$

Our RSUMM ensures that sentiment in a document is preserved to the maximum extent by optimizing X. Thus, we have a subjective excerpt of d, discarding objective sentences for

[9] An annotated collection of subjective and objective sentences.

effective sentiment classification. We use an n-gram model to represent the subjective excerpt of a document as a feature vector to the classifier.

4 Feature selection

RSUMM extracts most subjective sentences from a document. As sentences are relatively larger text units compared with words or phrases, using an n-gram model to convert them into feature vectors leads to a very high-dimensional feature set. For faster learning and better classification accuracies, we have to reduce this dimensionality by selecting features that are more relevant and capable of discriminating the class variable. Hence, a feature selection phase is essential in our case. We apply two state-of-the-art feature selection methods that are proven effective in text categorization and sentiment classification, mutual information (MI) and Fisher discriminant ratio (FDR), to select the final subjective features from a document (Yang and Pedersen 1997; Wang et al. 2009).

4.1 Mutual information

Mutual information (MI) is a widely used feature selection method in text categorization. It computes the mutual dependence between a feature f and a class C. It measures the amount of information that the presence/absence of a feature contributes to making the correct classification decision on C. In our case, the feature f is an n-gram, and class C is either positive or negative.

$$MI(f;C) = P(f,C)\log\left(\frac{P(f,C)}{P(f)P(C)}\right), \tag{8}$$

where $MI(f; C)$ denotes the mutual information between feature f and class C, $P(f, C)$ denotes the conditional probability of the feature occurring in class C, $P(f)$ denotes the probability of feature in the entire collection, and $P(C)$ denotes the class probability.

4.2 Fisher discriminant ratio

The Fisher discriminant ratio (FDR) is one of the effective approaches for dimensionality reduction in pattern recognition. The main idea of the FDR is that the points in a D-dimensional space are projected in such a way that there is maximum difference between the means and minimum variance within each class. For a two-class classification problem, the FDR can be computed as follows:

$$J(w) = \frac{|m_1 - m_2|^2}{S_1^2 + S_2^2}, \tag{9}$$

where m_i denotes a mean, S_i denotes the within-class variance, and $i = 1, 2$.

We modify the above equation in such a way that it computes the discriminating power of a feature f. Let $d_{P,i}(i = 1, 2\ldots, m)$ and $d_{N,j}(j = 1, 2\ldots, n)$ denote the ith positive document and jth negative document, respectively. We define two random variables $d_{P,i}(f)$ and $d_{N,j}(f)$ as in Wang et al. (2009).

$$d_{P,i}(f) = \begin{cases} 1 & \text{if } f \text{ occurs in } d_{P,i} \\ 0 & \text{otherwise} \end{cases}$$

$$d_{N,j}(f) = \begin{cases} 1 & \text{if } f \text{ occurs in } d_{N,j} \\ 0 & \text{otherwise} \end{cases}$$

The modified version of Eq. (9) is written as follows:

$$FDR(f) = \frac{\left(\frac{m_1}{m} - \frac{n_1}{n}\right)^2}{\frac{\sum_{i=1}^{m}(d_{P,i}(f) - \frac{m_1}{m})^2}{m} + \frac{\sum_{j=1}^{n}(d_{N,j}(f) - \frac{n_1}{n})^2}{n}}, \tag{10}$$

where m and n denote the total number of documents in class P and N, respectively, m_1 and n_1 denote the number of instances of feature f in class P and N, $d_{P,i}(f)$ denotes the presence or absence of feature f in review i of class P, and $d_{N,j}(f)$ denotes the presence or absence of feature f in review j of class N.

4.3 Final subset selection

In each case, high MI and FDR values of f imply that it has more discriminative power. We sort features in decreasing order of their corresponding MI and FDR values. In MI, we use a round-robin way of selecting features as we compute the mutual information of f with each class variable C. In FDR, the estimate is for the entire collection and not per class. We retain the top $Y\%$ of n-grams for each method in the final feature set.

5 Experiments

We conduct experiments on a movie review dataset collected from the IMDb archive. We use an n-gram model to represent the total review and its subjective excerpt as a feature vector to the classifier. Each n-gram is weighted using the *tfidf* score. We predict the overall polarity of a review as positive or negative.

5.1 Experimental setup

5.1.1 Datasets

We downloaded the available IMDb archive[10] of the rec.arts.movies.reviews newsgroup.[11] It contains 27,886 unprocessed and unlabeled HTML files that convey opinions of different authors on different movies. Predominantly, it has reviews rated on three different scales: 0–4, 0–5, and grade F to A+. A set of rules are framed to mine the rating patterns from the unprocessed set. Following these rules, we annotate each review as positive or negative.

With respect to the rating scale of 0–5, reviews having a rating of three and a half (***1/2) or above are annotated as positive. Reviews with a rating of two and a half (**1/2) and below are annotated as negative. With respect to the rating scale of 0–4, we annotate reviews with rating of three (***) or above as positive and reviews with rating of one and a half (*1/2) and below as negative. We annotate movie reviews that are graded B or above

[10] http://www.cs.cornell.edu/people/pabo/movie-review-data/polarity_html.zip.

[11] http://reviews.imdb.com/Reviews.

(B, B+, A−, A, A+) as positive and C− or below (C−, D, D−, F) as negative. Using this annotation scheme, we annotated 7,700 reviews as positive and 3,600 reviews as negative.

We have selected about 11,300 reviews from the archive for our experiments. This is a comparatively larger movie review dataset compared with others used in sentiment classification (Pang et al. 2002; Mullen and Collier 2004; Pang and Lee 2004; Matsumoto et al. 2005). We call this dataset polarity dataset (PDS) in the remainder of this paper. We split PDS into ten equal folds and perform tenfold cross-validation to test the statistical significance of our results. We maintain uniform class distribution across each fold.

We use a movie review dataset as a validation set to estimate parameters. It contains 1,400 reviews with equal class distribution and each review labeled as positive or negative. We call this as validation dataset (VDS)[12] in the remainder of this paper (Pang et al. 2002). To populate the knowledge of subjectivity into our experiments, we use an annotated collection of 5,000 subjective and objective sentences, respectively. We call this dataset as subjectivity dataset (SDS).[13] (Pang and Lee 2004)

5.1.2 Preprocessing

We extract body text from unprocessed HTML files in PDS. We have framed a set of rules to extract the body text. Since all unprocessed files are from the same source (IMDb), framing rule-based patterns for extraction is not very difficult, although we may have a little noise. We use the sentence breaker and the tokenizer implemented in the openNLP[14] tool to split the extracted body text into sentences, and sentences into words. Except for preprocessing, we do not use any linguistic resource in our experiments. In subjective feature extraction, we rely on statistical measures as described above. Hence, we reduce resource dependency or usage of complex patterns in subjective feature extraction.

5.1.3 Classifier and evaluation

We use the SVM classifier implemented in the SVMLight package[15] in our experiments, with parameters set to their default values. We focus more on extracting subjective features and representing them as feature vectors to the classifier rather than tuning its parameters.

We use the accuracy of the classifier on a test set as the evaluation metric. The accuracy of the classifier is calculated as shown in Eq. (11).

$$ACC = \frac{t}{n} \times 100, \tag{11}$$

where t denotes the number of samples correctly classified and n denotes the total number of test samples.

Since PDS is skewed towards the positive class, we modify the above equation slightly to include per-class accuracy to give the best generalization for our results. The modified equation for a two-class problem is shown in Eq. (12) (Baccianella et al. 2009).

$$ACC = \frac{\frac{t_P}{n_P} \times 100 + \frac{t_N}{n_N} \times 100}{n_C}, \tag{12}$$

[12] http://www.cs.cornell.edu/people/pabo/movie-review-data/mix20_rand700_tokens_cleaned.zip.

[13] http://www.cs.cornell.edu/people/pabo/movie-review-data/rotten_imdb.tar.gz.

[14] http://opennlp.sourceforge.net/.

[15] http://svmlight.joachims.org/.

where t_P denotes the number of test samples correctly classified as positive (P), n_P denotes the total number of test samples with label positive, t_N denotes the number of test samples correctly classified as negative (N), n_N denotes the total number of test samples with label negative, and n_C denotes the number of classes present in the dataset (in our case, $n_C = 2$).

5.2 Estimating parameters

We estimate optimal values for the parameters X and Y on VDS and later apply them on PDS. We use the dataset VDS as C_{pol} in Eq. (3) and SDS as C_{sub} in Eq. (4). We derive the corresponding term vectors $\overline{adf}$ and $\overline{asm}$, and use RSUMM to estimate the subjectivity of each sentence $\overline{s}$ in a review as described in Sect. 3.

We start with a very low value of $X = 10$, i.e., retaining the top 10% of sentences in each review as its subjective excerpt. We then incrementally add 5% in each iteration and examine the increase or decrease in the performance of the classifier. We use unigram representation of the subjective excerpt as a feature vector to the classifier. We perform tenfold cross-validation to test the statistical significance of our results, maintaining uniform class distribution in each fold. We choose the value of X that produces the best accuracy on VDS as its optimal value. The effect on accuracy of increasing X is depicted in Fig. 1.

From this figure, it is clear that, using an excerpt of a review, we are able to achieve better accuracies than using the total content. Using the total review, we are able to achieve accuracy of 76.1% with unigrams as features. However, retaining only 65% of it, the increase in the performance of the sentiment classifier is around 7%. This validates our assumption that the entire review cannot be subjective but rather is a mixture of subjective and objective information. We use the same value of 65 for X on PDS in our subsequent experiments.

For estimating Y, we start with a very high value of Y of 95, i.e., placing 95% of features in the final feature set for each method. We then, decrement Y by 5 for every iteration. The optimal value of Y for MI and FDR is the one that produces the best accuracy value with the respective n-gram model. The experimental results with unigrams ($n = 1$) as features

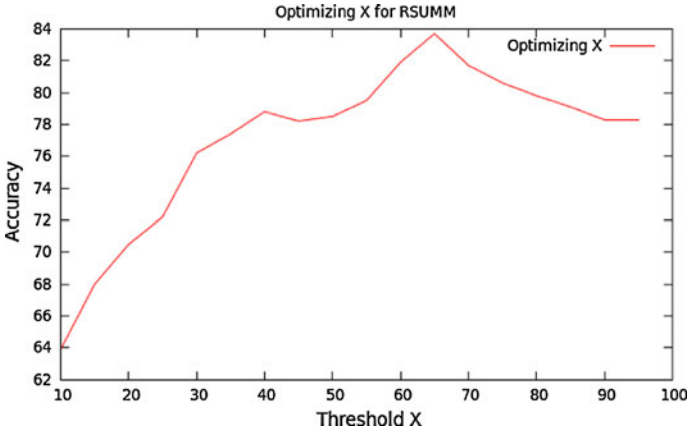

Fig. 1 Effect on accuracy of increasing X for RSUMM on VDS

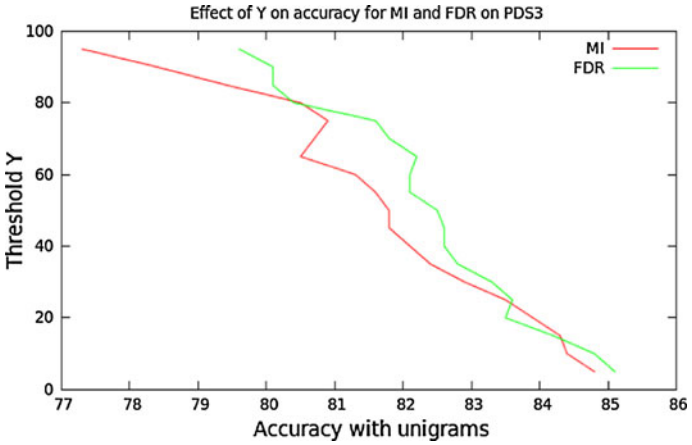

Fig. 2 Tuning the parameter Y for MI and FDR on RSUMM with unigrams as features

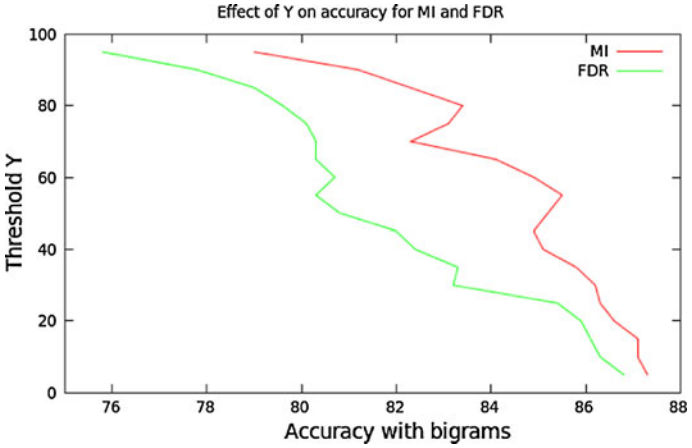

Fig. 3 Tuning the parameter Y for MI and FDR on RSUMM with bigrams as features

for MI and FDR on the total review are shown in Fig. 2. The effect on accuracy of changing Y for MI and FDR with bigrams ($n = 2$) as features is shown in Fig. 3.

From Figs. 2 and 3, it is evident that very low threshold values for Y produce accurate results in each method for a given n-gram model. The optimal value of Y in each case is 5. This conveys that very few n-grams in the entire text are indeed subjective. There is a lot of noise or misleading text surrounding the subjective features. We have to discard this for efficient sentiment classification, as proved in the literature (Pang and Lee 2004). In this work, we limit ourselves to using unigrams, bigrams, and their combination as feature vector representation of the text.

5.3 Experimental results

The baseline (BL) in our experiments is using the total review with unigrams (Uni), bigrams (Bi), and their combination (Uni+Bi) as features. We split each review into two

equal halves and carry out experiments using the top half (TH) and bottom half (BH) of the review. This is done to test the general pattern followed by authors in expressing their sentiment. The general pattern in movie review domain is that authors discuss objective information such as plot, characters, and other aspects of a movie at the beginning. They convey their sentiments at the end of a review (Pang and Lee 2004). We report the accuracy values of BL, TH, and BH on PDS, respectively, in Table 1.

We carried out two more experiments to test the relative significance of RSUMM. Firstly, we train a naive Bayes classifier on SDS and test it on each sentence of a review in PDS. We discard sentences labeled as objective by the naive Bayes classifier and retain subjective sentences in the subjective excerpt. We call this method of extracting subjectivity as default subjective excerpt (DSE). We represent the subjective excerpt using n-grams and then apply MI and FDR on the resultant n-grams to obtain the final feature set. Table 3 reports the accuracy values using DSE and the effect of applying MI and FDR on it. Secondly, we apply MI and FDR on the baseline to verify whether the performance of the sentiment classifier is more sensitive to Y than to X. The results for this experiment are reported in Table 2.

We use the collection PDS as C_{pol} in Eq. (3) and derive the respective $\overline{adf}$ term vector. The $\overline{asm}$ vector is derived using the collection SDS as C_{sub} in Eq. (5). After obtaining the corresponding $\overline{adf}$ and $\overline{asm}$ vectors, we compute the lexical similarity score between each sentence $\bar{s}$ of a review and $\overline{asm}, \overline{adf}$ vectors as per Eq. (6). We use a combination of two similarity scores to obtain the final subjective score of a sentence as shown in Eq. (7). We retain the top $X\%$ of sentences of a document in its subjective excerpt. We choose the value of X to be 65. We use an n-gram model ($n <= 2$) to represent the subjective excerpt as a feature vector to the classifier. We then apply MI and FDR on the resultant unigrams and

Table 1 Results showing CV accuracies for BL, TH, and BH on PDS

Features	BL	TH	BH
Uni	77.9	74.4	77.1
Bi	63.5	58.4	60.5
Uni+Bi	74.8	70.9	74.5

Table 2 Results showing CV accuracies for BL+MI and BL+FDR on PDS

Features	BL+MI	BL+FDR
Uni	80.4	81.7
Bi	75.6	74.6
Uni+Bi	79.9	80.4

Table 3 Results showing CV accuracies for BL+DSE, BL+DSE+MI, and BL+DSE+FDR on PDS

Features	BL+DSE	BL+DSE+MI	BL+DSE+FDR
Uni	79.4	80.8	81.5
Bi	64.2	80.1	79.0
Uni+Bi	77.3	81.4	81.2

Table 4 Results showing CV accuracies for BL+RSUMM, BL+RSUMM+MI, and BL+RSUMM+FDR on PDS

Features	BL+RSUMM	BL+RSUMM+MI	BL+RSUMM+FDR
Uni	83.9	86.1	**86.7**
Bi	69.3	83.9	**84.5**
Uni+Bi	80.1	**85.7**	84.9

bigrams with Y being 5 in each case to obtain the final feature set. The accuracy values for RSUMM, MI, and FDR are reported in Table 4.

6 Discussion

The baseline accuracy values reported in Table 1 indicate that unigram feature vector representation of a review yield better results compared with bigrams or combination of unigrams and bigrams. Using the total content with unigrams as features, the baseline accuracy reported is 77.9%. In case of bigrams and combination of unigrams and bigrams, the accuracy values are 63.5% and 74.8%, respectively. The bottom half of the review in the movie review domain is more subjective compared with the top half with each feature vector representation of the text. This conveys that authors model subjective expressions after discussing the plot, establishing the characters, etc. in the movie review domain. Using only the bottom half of each review with unigram representation of feature vectors, we were able to achieve accuracy comparable to the baseline of the sentiment classifier (77.9%–77.1%). In the case of bigrams there is a drop of about 3% in accuracy value from the baseline. However, the drop in accuracy is very small when the combination of unigrams and bigrams are used as features. This adds robustness to our assumption that the entire review does not contain subjective information.

Using only the top half of a review, there is a drop of about 3.5%, 5%, and 5% from the baseline in case of unigrams, bigrams, and their combination, respectively. There is a drop in the performance of the classifier when bigrams and combination of unigrams and bigrams are used as features rather than unigrams. This conveys that too many features will degrade the performance of the sentiment classifier and that there have to be selection criteria. Hence, researchers have defined some support thresholds in their experiments for selecting features (Pang et al. 2002; Matsumoto et al. 2005). From the results in Table 1 we conclude that unigram representation yields better results compared with bigrams or combination of unigrams and bigrams.

To test the relative significance of RSUMM, we use the DSE method to obtain the subjective excerpt of a document. The increase in performance of the classifier is around 1.5%, 0.7%, and 2.5% in case of unigrams, bigrams, and their combination, respectively. Using RSUMM, there is a significant increase in performance of the classifier from the baseline with unigrams as features (around 6%). In case of bigrams and combination of unigrams and bigrams, the increase is 5.8% and 5.3%, respectively. This conveys that RSUMM is able to optimize the essence of sentiment in a document with in an estimated threshold X. In comparison with DSE, RSUMM performed better, although they use the same SDS to populate the knowledge of subjectivity. This can be attributed to the fact that we use an additional metric called ADF in RSUMM that gives importance to the most informative sentences in a document. We do not use any linguistic resource in our experiments yet obtain an increase in performance of the sentiment classifier. These

experimental results prove that subjective feature extraction is possible with minimum use of linguistic resources and no complex patterns.

Feature selection techniques have proved vital in the performance of several text categorization tasks, as they enhance the performance of the classification system considerably (Yang and Pedersen 1997). Even in sentiment classification tasks, selecting features based on techniques such as document frequency, term frequency, minimum variance, etc. is done for obtaining good performance (Pang et al. 2002; Matsumoto et al. 2005; Baccianella et al. 2009). In our experiments, we employ two state-of-the-art feature selection methods, MI and FDR. From Tables 3 and 4, we can infer that there is little to choose between the two, as they both enhance the performance of the sentiment classifier. In case of unigrams, the increase in accuracy of RSUMM+FDR is less comparable to RSUMM (2.8%–6.0%). In case of bigrams, the performance of the classifier is highly sensitive to the parameter Y. This implies the presence of a large number of irrelevant features in the documents.

Using MI as the feature selection method, there is a marginal increase of about 2% from baseline in case of unigrams, whereas there is a significant increase of about 12% in accuracy value for bigrams. After applying MI with RSUMM, there is an increase of 2.2% and 14% with unigrams and bigrams, respectively. Using the combination of DSE and MI, there is an increase of 1.4%, 16%, and 4% in case of unigrams, bigrams, and their combination, respectively. Using RSUMM and MI, the accuracy values obtained for unigrams, bigrams, and their combination are 86.1%, 83.9%, and 85.7%, respectively. The increase in their values from the baseline is about 8%, 20%, and 11% with unigrams, bigrams, and their combination as features, respectively, which is significant.

FDR as the feature selection technique performed very similar to MI. In case of unigrams, it increased the performance of the classifier compared with MI from the baseline (81.7%–80.4%). Using the combination of RSUMM and FDR, we obtain the highest accuracy of 86.7% and 84.5% with unigrams and bigrams, respectively. There is an increase in the accuracy values of about 9%, 21%, and 10% for each feature representation, respectively, from the baseline. There is a slight drop in the accuracy values when the combination unigrams and bigrams are used as features compared with unigrams and bigrams in isolation.

7 Conclusions and future work

We focused on subjective feature extraction, the key component in document sentiment classification. We followed a two-step filtering methodology to mine subjective portions in a document. We used RSUMM to obtain the subjective excerpt of a document by estimating subjectivity at the sentence level. We then applied well-known feature selection methods on the subjective excerpt to obtain the final feature set. The major contributions of this work are:

1. We explored various subjective feature extraction methodologies in sentiment classification and their limitations.
2. We attempted to minimize resource dependency or complex patterns in subjective feature extraction.
3. We developed a simple and statistical methodology called RSUMM to extract subjectivity in a document.

We explored several resource-independent or language-independent subjective feature extraction methods from the literature. We used techniques similar to the vector space model in RSUMM to estimate the subjectivity of each sentence. To the best of our knowledge, use of techniques similar to the vector space model for extracting subjective features has not yet been proposed in the literature. Based on our experimental results, we conclude that subjective feature extraction is possible with minimum use of linguistic resources. We explored two frequency-based metrics (ADF and ASM) and used them in RSUMM to estimate subjectivity. We then applied two well-known feature selection methods (MI and FDR) on the subjective excerpt of RSUMM.

Our work can be considered as a building block for analyzing sentiment with minimal usage of linguistic resources and no complex patterns. In the future, we need to explore different metrics to extract subjectivity, and conduct experiments. As feature selection methods have proved critical in the performance of classification, we need to explore more novel methods for selecting features.

Acknowledgments We thank Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan for providing the URL in their paper to download the IMDb movie review dataset. We thank the Department of Computer Science, Cornell University for providing the link to download the dump of the IMDb archive.

References

Argamon, S., Koppel, M., & Avneri, G. (1998). Routing documents according to style. In *Proceedings of 1st international workshop on innovative information systems*.

Aue, A., & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of the international conference RANLP-2005*.

Baccianella, S., Esuli, A., & Sebastiani, F. (2009). Multi-facet rating of product reviews. In *Proceedings of the 31th European conference on IR research on advances in information retrieval*, ECIR '09 (pp. 461–472). Springer-Verlag.

Baeza-Yates, R. A., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Boston: Addison-Wesley Longman

Beineke, P., Hastie, T., & Vaithyanathan, S. (2004). The sentimental factor: Improving review classification via human-provided information. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, ACL '04. Association for Computational Linguistics.

Cui, H., Mittal, V., & Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *Proceedings of the 21st national conference on artificial intelligence* (Vol. 2, pp. 1265–1270). AAAI Press.

Gretzel, U., & Yoo, K. H. (2008). Use and impact of online travel reviews. *Information and Communication Technologies in Tourism* (pp. 35–46).

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 8th conference on European chapter of the association for computational linguistics* (pp. 174–181). Association for Computational Linguistics.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 168–177). KDD '04.

Hu, Y., Lu, R., Li, X., Chen, Y., & Duan, J. (2007). A language modeling approach to sentiment analysis. In *Proceedings of the 7th international conference on computational science, Part II*, ICCS '07 (pp. 1186–1193). Springer-Verlag.

Kessler, B., Numberg, G., & Schütze, H. (1997). Automatic detection of text genre. In *Proceedings of the 35th annual meeting of the association for computational linguistics and 8th conference of the European chapter of the association for computational linguistics*, ACL-35 (pp. 32–38). Association for Computational Linguistics.

Li, S., Lee, S. Y. M., Chen, Y., Huang, C.-R., & Zhou, G. (2010). Sentiment classification and polarity shifting. In *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)* (pp. 635–643).

Liu, B. (2010). Sentiment analysis and subjectivity. In *Handbook of natural language processing* (2nd ed.). Boca Raton, FL: CRC Press, Taylor and Francis Group.

Matsumoto, S., Takamura, H., & Okumura, M. (2005). Sentiment classification using word sub-sequences and dependency sub-trees. In *Proceedings of PAKDD* (pp. 301–311).

Mullen, T., & Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP* (pp. 412–418).

Pang, B., & Lee, L. (2004) A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL* (pp. 271–278).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundation and trends in information retrieval, 2*, 1–135. ISSN 1554-0669.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on empirical methods in natural language processing* (Vol. 10, pp. 79–86).

Raychev, V., & Nakov, P. (2009). Language-independent sentiment analysis using subjectivity and positional information. In *Proceedings of the international conference RANLP-2009* (pp. 360–364). Association for Computational Linguistics.

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009). Adapting naive bayes to domain adaptation for sentiment analysis. In *Proceedings of the 31th European conference on IR research on advances in information retrieval*, ECIR '09 (pp. 337–349). Springer-Verlag.

Thet, T. T., Na, J.-C., & Khoo, C. S. (2008). Sentiment classification of movie reviews using multiple perspectives. In *Proceedings of the 11th international conference on Asian digital libraries: Universal and ubiquitous access to information*, ICADL 08 (pp. 184–193). Springer-Verlag.

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, ACL '02 (pp. 417–424). Association for Computational Linguistics.

Wang, S., Li, D., Wei, Y., & Li, H. (2009). A feature selection method based on fisher's discriminant ratio for text sentiment classification. In *Proceedings of the international conference on web information systems and mining*, WISM '09 (pp. 88–97). Springer-Verlag.

Whitelaw, C., Garg, N., & Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on information and knowledge management*, CIKM '05 (pp. 625–631). ACM.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the 14th international conference on machine learning*, ICML '97 (pp. 412–420). Morgan Kaufmann